**Summaries**

[1], Adversarial Machine learning technique used to deceive the Deep Learning (DL) based NIDS that made DL-NIDS a vulnerable. So, instead of securing the network, the network will become more vulnerable due to deep learning infrastructure. There are two methods to secure deep learning models i.e. reactive and proactive. Defensive distillation and adversarial training are two proactive techniques while Pixel Defense is an example of reactive defense. An Open Source library Cleverhans is available for adversarial training. False positives detection increases the reliability of system while false negatives show the efficiency of system. There is a tradeoff between both of these values. There are two adversarial attacking objectives i.e. integrity violation exploited by false negative and availability violations exploited by false positive.

[2], NIDS are basically of 2 types: Signature based NIDS (SNIDS) and Anomaly Detection based NIDS (ADNIDS). SNIDS used for detecting the intrusion based on known features while ADNIDS is used for detecting unknown featured intrusions. In this paper, Self-Taught Learning (STL) and Non-symmetric Deep Auto-Encoder(NSDA) is used to learn patterns in unsupervised learning. Sparse auto-encoder will be used to learn the unknown patterns and then logistic regression function will be used to classify the user behavior learned by stack autoencoder. There are already many NIDS researched and implemented but a real time and more generic NIDS requires a lot of work yet. Total 115 features are considered here that are given as input to neural network. Deep network is made by stacking the autoencoders and logistic regression function is used to classify user as an intruder or normal user.

[3],This paper presents a novel deep learning technique for intrusion detection. Nonsymmetric deep autoencoder (NDAE) for unsupervised feature learning is proposed in this paper. This has been implemented using GPU enabled Tensor Flow with datasets KDD Cup ‘99 and NSL-KDD datasets. Most solutions are signature-based rather than anomaly based. Where signature based solutions are those solutions that detect an attack which is known .i.e. it belongs to a library/dataset of known attacks. On the other hand Behaviour or anomaly based as the name suggest is based upon the pattern of normal behaviour .i.e whenever an odd behaviour over a machine is observed the indication is made. Cetainly , the volume and diversity of network data is a challenge. However, here deep and shallow learning techniques are deployed. Particularly, Random Forest and NDAE are used. This paper focuses on nonsymetric data dimentionality reduction. For data dimensionality reduction the technique used is auto encoder. The input is first transformed into a typically lower-dimensional space (encoder), and then expanded to reproduce the initial data (decoder). Once a layer is trained, its code is fed to the next, to better model highly non-linear dependencies in the input. This paradigm focuses on reducing the dimensionality of input data. To achieve this, there is a special layer - the code layer , at the centre of the deep auto-encoder structure. This code layer is used as a compressed feature vector for classification or for combination within a stacked auto-encoder. If applied deeplearning to autoencoders it is termed as deep auto encoders or Stacked auto encoders. As for this paper NDAE Non symmetric deep auto encoder is used. Which learns the important features using similar strategy to that of deep auto encoder. Hidden input vector is given by hi = σ(Wi . hi−1 + bi) and “sigmoid function” is used for computing the actual output. To learn more complex features Stacked NDAE may be used. The stacked auto encoders with soft-max layer is not good at classification as compared to other classification algorithms so here stacked NDAE with shallow learning classifier is opted, .i.e Random Forest Classifier Algorithm.

[4], this paper proposed Anomaly-based Network Intrusion Detection System (A-NIDS) using Long Short Term Memory (LSTM) neural network architecture mostly used for sequence prediction problems, and One-class classification method Support Vector Data Description (SVDD) used for building a good data description vectors for normal data. This approach is used to tackle the time sequence attacks i.e. Dos, Probing, R2L and U2R etc. SVDD provides a best way to detect any anomalous activity because it generates a hyper sphere shape boundary across normal traffic data, and then classify any outlier data as anomalous. Network packets are fed to LSTM in form of sequence and the output of each sequence is averaged using mean pooling method. The LSTM model parameters are initialized randomly and then adjusted during training. Finally the output vectors of all sequences are fed to SVDD model, that convert them in a comparable form i.e. support vector classifier (SVC) and then make a hyper sphere shape space that will distinguish normal data from anomalous data. The new sequences will be input to SVDD Data Description model to determine a classification function y that will classify the sequence as normal or anomalous. For training, first order gradient descent approach is used due to its better results and high performance. In this way, parameters (c, R, theta) are optimized by joint structure of LSTM and SVDD infrastructure.

[5], this paper proposed that

[6], This is actually a final year project dicussed by one of the students who have been part of it . So it starts with highlighting the fact that IDS is a variable sort of thing rather than static. Hence, approach used here is to detect anomaly. So it uses ISCX dataset by Canadian University for Cyber Security. An important step of Data pre-processing that enables “Deep learning” to be applied is that ISCX Flowmeter software has been used to convert “.PCAP” files to “.xml” format and finally to NumPy arrays. Essentially, these arrays act as images when given a visual form. Final product is an image, it makes sense that Convolutional Neural Network approach can be applied or generally Deep Learning. So VGG-19 model was used for this purpose. However, firstly anomaly dataset and Normal dataset individually were used for IDS as training data but there were discrepencies in the results. But when both Anomaly Data and Normal data were used the results improved substantially. And it became more like an outlier detection problem or in simple words classification problem. The system worked fine and to cope with various new types of attacks updating the dataset periodically ,as the new dataset by Canadian University for Cyber Securirty is launched makes system more efficient.

[7], As in most papers on this topic all start with mentioning the two types of Intrusion Detection techniques. That are 1. Signature Based 2. Anomaly Detection. Signature Based follow similar technique to that of a Virus Scanner. And as you know Virus Scanner needs to be updated periodically so is Signature based NIDS which require updating the datasets. While Anomaly Based works on analyzing the pattern of usage. Its ability to detect previously unknown (or variants of known) attacks when they appear is the biggest advantage of an Anomaly-based System. Self-Taught Learning is a deeplearning approach that consists of two stages for classification. In first stage un labeled data is used for feature learning i.e Unsupervised Feature Learning and this learnt representation in second step is applied to labelled data to carry out classification. The Sparse Auto-Encoder based feature learning is used for this work due to its easy implementation and good performance. As for dataset NSL KDD is used. One of the major drawback with the KDD Cup dataset is that it contains an enormous amount of redundant records both in the training and test data. This redundancy makes the algorithms biased towards frequent attacks and as a result does poor classification of infrequent harmful records. On the contrary NSL KDD dataset was more improved in a sense that it was less redundant hence making algorithms giving more accurate results. The dataset is preprocessed before applying self-taught learning on it. Nominal attributes are converted into discrete attributes using 1-to-n encoding The values in the output layer are computed by Sigmoid Function which gives values between 0 and 1 and then max min operations are performed to get a new list of attributes. The NSL-KDD training data without labels is used with the new attributes for the feature learning using sparse auto-encoder for the first stage of self-taught learning. In the second stage, the new learned features representation is applied on the training data itself for the classification using soft-max regression. In this implementation, both the unlabeled and labeled data for feature learning and classifier training come from the same source, i.e., NSL-KDD training data.

[8], So this report was important because we get to see how different ways can be used for NIDS and all through Deep Learning. 1.Vanilla Deep Neural Net 2. Self Taught Learning 3.Recurrent Neural Network are the three models of deep learning that primrarily are discussed in this report. The attacks which are on top of the list are DoS, Probe, R2L and U2R. KDD and NSL KDD datasets were used to observe difference in performances. Over 7 weeks’ network traffic collected was used for training. In all this a very important point is that “most novel attack are derived from known attacks” so it pretty much makes sense that deep learning can be applied for NIDS and to get effective results. Again as in one of the papers that I have researched on, here the fact is highlighted that KDD NSL dataset was prefered over KDD Cup due redundancies in the latter one. Simple deep neural network attained accuracy of 66 percent. It well classified the DoS and probe attacks but there was less success classifying normal non-threatning requests and U2R attacks. Then comes the STL(self taught learning). In steps STL works by first converting categorical features to numeric values. Then min max normalization takes place. Feature vector is passed to two layered stacked auto encoder. Results with this approach were pleasing with accuracy of 98.9 percent. Last but not the least Recurrent Neural Network take the input along with the previously perceived input instance. Problem with this model was vanishing gradient problem(when gradient is very small and weights remain same). Therefore the Long Short Term Memory Networks is a better version of RNN which eliminates the issue of vanishing gradient problem. Accuracy observed was 79.2 percent. Importantly the model failed to predict attacks other than DoS.

[9], This paper starts by discussing the Network and Network Intrusion Detection Systems. So amongst the existing system the Cross Layer Design is defined as: “Cross layer design, where the boundary among the protocol layers is a violated by sharing internal information, helping layers to become aware of the changes in the others and provide higher quality of services to the user.” Furthermore, the Network layers are discussed. So, important part is implementation. During the data transmission, packet drop process and packet delaying can be occurred due to the congestion and the collision along with the unavailability of the channel beau case of the hidden terminal and exposed terminal problem. This will lead to the false detection of the normal behavior as malicious behavior in the network environment. In order to handle this network dynamics, the learning system is employed based on the reinforcement learning system. So, behavioral prediction and operation of decision making system is carried by collected input during learning process. Secondly, the collaborated view of hidden layer is devised by **Bayesian network with Boltzmann input**. it is complex task to train a Deep Boltzmann machine (DBM) with **approximate maximum-likelihood learning** using the stochastic gradient unlike the restricted Boltzmann machines (RBM). DBM is a recently introduced variant of Boltzmann machines which extends widely used RBM to a model that has multiple hidden layers. There are some pros of DBM. DBM’s have the potential of learning internal representations that become increasingly complex, which is considered to be a promising way of solving object and speech recognition problems. Also high-level representations can be built from a large supply of unlabeled sensory inputs and very limited labeled data can then be used to only slightly fine-tune the model for a specific task at hand. As far as results are concerned, In case of Key Generation Delay, Machine **Learning Software Defined Network (MLSDN)** achieves higher performance by obtaining the lower delay. Similarly for **Key SharingDelay** MLSDN in lower latency compare to Software Defined Network (SDN). In case of **Hash Generation Delay**, MLSDN achieves lower delay while generating the hash code.

**References:**

[1] 27 March,2019. Link: <https://arxiv.org/pdf/1903.11688.pdf>

[2] March 2019. Link: <https://search.proquest.com/openview/ab6d5a2cbbb23e2c3cc10af0093a3134/1?pq-origsite=gscholar&cbl=2026671>

[3] February 2018: <https://blogs.qub.ac.uk/netsec/wp-content/uploads/sites/153/2019/02/08264962.pdf>

[4] May 2019: <https://arxiv.org/ftp/arxiv/papers/1906/1906.05277.pdf>

[5] 3rd April, 2019: <https://ieeexplore.ieee.org/abstract/document/8681044>

[6] Building Intrusion Detection System using Deep Learning year: 2018

link: <https://towardsdatascience.com/building-an-intrusion-detection-system-using-deep-learning-b9488332b321>

[7], Detecting Network Intrusion through a Deep Learning Approach:2018

Link:<https://www.researchgate.net/publication/322572874_Detecting_Network_Intrusion_through_a_Deep_Learning_Approach>

[8], Comparative Study of Deep Learning Models for Network Intrusion Detection:2018.

url: <https://scholar.smu.edu/datasciencereview/vol1/iss1/8/?%20utm_source=scholar.smu.edu%2Fdatasciencereview%20%2Fvol1%2Fiss1%2F8&utm_medium=PDF&utm_campaign=PDFCoverPages>

[9], Optimized Intrusion Detection System using Deep Learning Algorithm:2019

Link: <https://www.academia.edu/39171572/Optimized_Intrusion_Detection_System_using_Deep_Learning_Algorithm>

<https://arxiv.org/pdf/1511.07275.pdf>